1996, entitled DATE FORMATTING AND SORTING FOR DATES SPANNING THE TURN OF THE CENTURY ("the patent"). A first reexamination was Ordered by the USPTO itself (005,592). A second Reexamination was Ordered in response to a Petition of a first anonymous Requestor (005,628). A third reexamination was Ordered in response to a Petition of a second anonymous Requestor (005,727).

Claims 1-76 were in the above captioned cases. Claims 1-76 have been rejected. Claims 1-76 remain in this case. Claim 54 has been amended, solely for the purpose of correcting a typographical error in the claim, namely that the word "representations" was left out of the claim as filed. For the below stated reasons Applicant respectfully traverses the Examiner's rejections of claims 1-76 and asserts that claims 1-76 should be allowed and respectfully requests that the Examiner allow claims 1-76.

As a preliminary matter, the examiner has referred to the following formalities:

> Objection to Oath/Declaration
>
> 1. The reissue oath/declaration filed with this application is defective because it fails to identify at least one error which is relied upon to support the reissue application. See 37 CFR 1. 175(a)(1) and MPEP §
> 1414.
>
> a. The declaration merely provides a statement of "inadequate claiming" as an error for the instant reissue.
> The "inadequate claiming" statement does not meet the requirement, as provided in MPEP § 1414.
>
> b. A reissue declaration must be signed by the inventor when the claims are being broadened. Although Mr. Dickens signed the present declaration as the assignee's representative, the inventor's averments provided herein are deemed to be adequate to satisfy the inventor's signature requirement. Consequently, the inventor's declaration provided herein is hereby accepted as the reissue declaration.
>
> c. The Oath/Declaration fails to identify the inventor's full name and citizenship as required by 35 U. S. C. 115 and 3 7 CFR 1.63 (a)(3).
>
> ASSENT OF ASSIGNEE UNACCEPTABLE
>
> ✓ 2. This application is objected to under 37 CFR 1. 172(a) as the assignee has not established its ownership interest in the patent for which reissue is being requested. An assignee must establish its ownership interest *in order to support the consent to a reissue application as required by 37 CFR* 1.172(a). The submission establishing the ownership interest of the assignee is informal. There is no indication of record that the party who signed the submission is an appropriate party to sign on behalf of the assignee. 37 CFR 3.73(b). Further, the establishment of ownership under 37 CFR 3.73 is insufficient since it relies upon certain attached documents, which have not been supplied to the office.

Consequently, a proper submission establishing ownership interest in the patent, pursuant to 37 CFR 1. 172(a), is required in response to this action.
3. This application is objected to under 37 CFR 1. 172(a) as lacking the written consent of all assignees owning an undivided interest in the patent. The consent of the assignee must be in compliance with 37 CFR 1. 172. See MPEP §1410. 01. A proper assent of the assignee in compliance with 3 7 CFR 1.172 and 3.73 is required in reply to this Office action.
4. The original patent, or an affidavit or declaration as to loss or inaccessibility of the original patent, must be received before this reissue application can be allowed. See 37 CFR 1.178.
5. Changes made in the certificate of correction have not been incorporated into the specification of the reissue application. Applicant is required submit a substitute specification which complies with reissue practice.
6. Claims 1-76 are- rejected as being based upon a defective reissue declaration under 35 U.S.C. 251 as set forth above. See 37 CFR 1. 175. The nature of the defect(s) in the declaration is set forth in the discussion above in this Office action.

Suggestions and Recommendations

6a. Regarding paragraph 1, the applicant's attention is directed to MPEP 1414, stating that: In identifying the error, it is sufficient that the reissue oath/declaration identify a single word, phrase, or expression in the specification or in an original claim, and how it renders the original patent wholly or partly inoperative or invalid."
6b. Regarding paragraphs 2-6, the applicant is advised to submit copies of the records evidencing the chain of assignment. The declaration should also be signed by the inventor acting as inventor in addition to acting on behalf of the assignee. Further, the statement under 37 CFR 3.73 should identify the relationship of the inventor to the assignee upon which the inventor relies to authorize his signature as one in which he is authorized to act on behalf of the assignee (i.e. what is his corporate office in the assignee.) Models of Consent by the assignee can be found on the USPTO web site at: http://www.uspto.eb/forms/sb0053.pdf
Applicant submits along with this Response a Reissue Declaration signed by Bruce M. Dickens as the inventor, including the statement of the claim language that renders the patent wholly of partly inoperative or invalid. Also submitted is a Consent of Assignee signed on behalf of the Assignee of record, Dickens-Soeder2000, LLC, by Bruce M. Dickens an Organizer and Member/Manager of the Assignee of record and authorized to sign on behalf of the Assignee of record.

Applicant submits along with this Response a Supplemental Certificate under 37 C.F.R. §3.73, along with the supporting assignment documentation and supporting documentation showing that the inventor Bruce M. Dickens is an Organizer of dickens-

Soeder2000 LLC and a declaration that he is authorized to sign the Certification on behalf of Dickens-Soeder2000.

Applicant respectfully requests that the Examiner withdraw the objections to the Application based upon 37 C.F.R. §1,172(a) and the rejection of claims 1-76 based upon 35 U.S.C. §251 and 37 C.F.R. §1.175.

Applicable Law

A claim is anticipated under 35 U.S.C. §102 only if each and every element as set forth in the claim is found, either expressly or inherently disclosed, in a single prior art reference. *Verdegaal Brothers v. Union Oil Co. of California*, 814 F.2d 628, 631, 2 U.S.P.Q.2d 1051,1053 (Fed. Cir. 1987). The identical invention must be shown in as complete detail as contained in the claimed invention. *Richardson v. Suzuki Motors Co.*, 868 F.1226, 1236, 9 U.S.P.Q.2d 1913, 1920 (Fed. Cir. 1989). The elements must be arranged as required by the claim. *In re Bond*, 910 F.2d 831, 15 U.S.P.Q.2d 1566 (Fed. Cir. 1990).

A prior art reference may be considered to teach away when " a person of ordinary skill, upon reading the reference, would be discouraged from following the path set out in the reference, or would be led in a direction divergent from the path that was taken by the [inventor]." *In re Gurley*, 27 F.3d 551, 553, 31 U.S.P.Q.2d 1130, 1131 (Fed. Cir. 1994). Simply because the claimed invention employs a known technique, i.e., windowing, does not, of itself, make the process of the claimed invention unpatentable. *In re Brower*, 77 F.3d 422, 37 U.S.P.Q.2d 1663 (Fed. Cir. 1994).

The obviousness test under §103 must be applied to the subject matter of the claimed invention as a whole. There is no "gist" or "heart" or "core" of the invention evaluated for obviousness purposes. It is necessary to consider all of the subject matter of the claimed invention. *Loctite Corp. v. Ultraseal Ltd.*, 781 F.2d 861, 228 U.S.P.Q. 90 (Fed. Cir. 1985). Even if the prior art references taken together show all of the elements of the claimed invention, it must still be shown that the claimed invention would have been obvious as a whole and it is improper to analyze the claimed invention by its separate parts, even if each is shown in the art. *Custom Accessories, Inc. v. Jeffrey-Allan Industries, Inc.*, 807 F. 2d 955, 1 U.S.P.Q.2d 1197 (Fed. Cir. 1986), and *In re Wright*, 848 F.2d 1216, 6 U.S.P.Q.2d 1959 (Fed. Cir. 1988). In the present case, neither Ohms, Hazama, Booth nor Shaughnessy, nor any combination thereof, discloses all of the elements of the claimed invention, and, therefore, there does not even exist support for a *prima facie* case for obviousness. *See*, MPEP § 2143.03, *In re Royka*, 490 F.2d 981, 180 U.S.P.Q. 580 (C.C.P.A 1974), and *In re Wilson*, 424 F.2d 1382, 1385, 165 F.2d 494, 496 (C.C.P.A. 1970). This alone is a basis for finding the claimed invention patentable over these references.

Hindsight must be avoided in combining references in the prior art. It is error to reconstruct the claimed invention from the prior art using the claimed invention as a blueprint. *Panduit Corp. v. Dennison Manufacturing Co.*, 774 F.2d 1082, 227 U.S.P.Q. 337, 343 (Fed. Cir. 1985), *In re Find*, 837 F.2d 1071, 5 U.S.P.Q.2d 1596 (Fed. Cir. 1988), and *W.L. Gore & Associates, Inc. v. Garlock, Inc.*, 721 F.2d 1540, 220 U.S.P.Q. 303 (Fed. Cir. 1983), *cert. denied*, 469 U.S. 851 (1984). It is improper to reject a claim based upon the mere assertion that one of ordinary skill in the art would know to add a claimed feature to the claimed invention without the citation of a reference that teaches the claimed feature (MPEP §706.02(a)), or at least an affidavit from the Examiner detailing the Examiner's knowledge of the prior art under 37 C.F.R. §1.107(b). *In re Newell*, 13 U.S.P.Q.2d 1248 (Fed. Cir. 1989), and *In re Kaplan*, 229 U.S.P.Q. 678, 683 (Fed. Cir. 1986).

A reference must be considered as a whole. A prior art reference that describes a product or process similar to the claimed product or process and also a statement that the product or process does not work does not teach the claimed product or process. A prior art reference may be considered to teach away when " a person of ordinary skill, upon reading the reference, would be discouraged from following the path set out in the reference, or would be led in a direction divergent from the path that was taken by the [inventor]." *In re Gurley*, 27 F.3d 551, 553, 31 U.S.P.Q.2d 1130, 1131 (Fed. Cir. 1994).

The Examiner's Rejections

Claims 1-3, 5, 7, 9-10 stand rejected under 35 U.S.C. §102(e) as anticipated by Daniel P. Shaughnessy, US. Patent No. 5,630,118, filed on November 21, 1994 and issued on May 13, 1997 (Shaughnessy, hereinafter) or, in the alternative, under 35 U.S.C.§103 (a) as obvious over Shaughnessy in view of Masakazu Hazama, Japanese Application No. 05-027947, published on February 5, 1993 (Hazama, hereinafter).

Claims 1-3, 5, 7, 9-10 stand rejected under 35 U.S.C.§103 (a) as being unpatentable over B. *G. Ohms, Computer Processing of Dates Outside the Twentieth Century,* IBM Systems. Journal, Volume 25, Number 2, 1986, pages-244-51, (Ohms, hereinafter), in view of Hazama.

The Examiner has taken the position that:

> As to claim 1, Shaughnessy discloses the claimed 'method of processing symbolic representations of dates stored in a database' as a method and system for modifying and operating a computer system to perform operations on date fields having a two digit representation for the year without mistaking the year 2000 and the year 1900 (col. 1, lines 7-14 et seq). In particular, Shaughnessy discloses the step of 'providing a database with symbolic representations of dates stored therein according to a format wherein M.sub.1 M.sub.2 is the numerical month designator, D.sub.1 D.sub.2 is the numerical day designator, and Y.sub.1 Y.sub.2 is the numerical year designator' as a database having a 6 digit-field for storing a Date type in the form of MMDDYY (see appendix in col. 18, Date Type A), wherein the MM represents the month, DD represents the day and YY represents the year for a particular six digit-date. Further, Shaughnessy discloses the claimed 'all of the symbolic representations of dates falling within a 10-decade period of time' as a date having a cycle or a range of a 100 years (col. 18, Cycle/Range C1 = THE DATE CYCLE IS 100 YEARS). As pointed out in column 2, lines 11 - 14 and column 3, lines 4-8 of Patent No. 5, 806,063, all dates in commercial and industrial databases span within one 100 year. Shaughnessy's system being of the commercial or industrial kind described in the cited patent, must therefore, as a practical matter, inherently incorporate this limitation. Shaughnessy also discloses the claimed step of 'selecting a 10-decade window with a Y.sub.A Y.sub.B value for the first decade of the window, Y.sub.A Y.sub.B being no later than the earliest Y.sub.1. Y.sub.2 year designator in the database' as a subroutine for determining the current date[1] to thereby select a 100 year cycle wherein the current date is the pivot date and wherein the cycle ends a 100 years from said current date (col. 5, lines 31-36). Additionally, Shaughnessy discloses the step of determining a century designator C.sub.1 C.sub.2 for each symbolic representation of a date in the database, C.sub.1 C.sub.2 having a first value if Y.sub.1 Y.sub.2 is less than Y.sub.A Y.sub.B and having a second value if Y.sub.1 Y.sub.2 is equal to or greater than Y.sub.A Y.sub.B' as the comparison of the current date to the date when the system was installed with the modifications (modified system install date) to thereby determine the century value (col. 5, lines 36-65 et seq). Alternatively, Shaughnessy discloses the comparison of the YYMMDD portion[2] of the date to the corresponding date portion at the end of the

[1] **The current date, by virtue of being the pivot date in the 100 year-cycle and by being initially set to the operating system date, initially set to 0000 (see col. 7, lines 19-20 et seq), is therefore not any later than the earliest date in 100 year-cycle in the database.**

[2] Shaughnessy specifically suggests that it ***might be*** desirable to set the current date to a date which

100 year cycle to thereby determine the century value (col. 7, lines 7-15 et seq). Finally, Shaughnessy discloses the step of 'reformatting the symbolic representation of the date with the values C.sub.1 C.sub.2, Y.sub.1 Y.sub.2, M.sub.1 M.sub.2, and D.sub.1 D.sub.2 to facilitate further processing of the dates' by appending the determined century value before the YYMMDD date in order to yield a CCYYMNMDD date format (col.5, lines 46-51; col.6, lines 57-65 et seq).

In the event that inherency fails to support the assertion that Shaughnessy discloses the limitation that the two digit date is smaller or equal to the smallest date in the database, where all the dates in the database fall within a 100 year period, it would have been obvious to the ordinary skilled artisan to look to the teachings of Hazama to complement Shaughnessy's.

Hazama discloses an analogous date processing system wherein, for a 100 year window (restricting the all the dates in the database between the 20th and 21st centuries), the pivot date for the window is selected based on a two digit year date that is smaller than the smallest two digit year date in the database (e.g. if smallest two digit date stored in database is 73, the pivot date for the 100 year window is -chosen to be 72). (page 4 of translated document, last paragraph). It would have been obvious to one of ordinary skill in the art of data processing to combine the teachings of the cited references. The ordinary skilled artisan having read Shaughnessy would immediately see the need to determine which 100 year span to use. This determination would, have led the ordinary skilled artisan to the Hazama reference, which teaches the pivot date being smaller than the smallest two digit date in the database having all the dates within a 100 year period as a solution to restrict the selection of Shaughnessy's window and thereby forcing all dates already stored in the database to fall in the 20th century.

As to claim 2, Shaughnessy disclose s the invention as discussed in the rejection of claim 1, as well as the claimed limitation whereby a '10-decade window includes the decade beginning in the year 2000' by suggesting the use of a 100 year window that includes a decade date in the 2 1 st century (col. 6, lines 28-29 et seq).

As to claim 3, Shaughnessy discloses the invention as discussed in the rejection of claim 2, as well as the claimed limitation whereby, 'the step of determining includes the step of determining the first value as 20 and the second value as 19' by assigning the century value to 19 if the YYDDD portion of the date is greater than or equal to the corresponding portion of the corresponding portion of the modified system install date (col. 5, lines 40-46) and by assigning the century value to 20 if the pivot date is less than the modified system install date (col.5, lines 52-60 et seq).

As to claim 5, Shaughnessy discloses the invention as discussed in the rejection of claim 1, as well as the claimed limitation, wherein 'the step of reformatting includes the step of reformatting each symbolic representation of a date into the format C.sub.1 C.sub.2 Y.sub.1 Y.sub.2 M.sub.1 M.sub.2 D.sub.I D.sub.2' as the conversion of the current date from a six-digit format (YYMMDD) into an 8-digit format (CCYYMMDD) (col. 5, lines 48-50 et seq).

compares low to all other dates (col. 7, lines 16-17 et seq).

As to claim 7, Shaughnessy discloses the invention as discussed in the rejection of claim 1, as well as the claimed limitation, wherein the step of providing a database includes the step of converting pre-existing date information having a different format into the format wherein M.sub.1 M.sub.2 is the numerical month designator, D.sub.1 D.sub.2 is the numerical day designator and Y.sub.1 Y.sub2 is the numerical year designator' by as the converting the current date in a six digit format (YYMMDD), wherein YY represents the year, MM represents the month and DD represents the day (col. 8, lines 18-27 et seq).

As to claim 9, Shaughnessy discloses the invention as discussed in the rejection of claim 1, as well as the claimed limitation of 'storing the symbolic representation of dates and their associated information back into the database after the step of reformatting' by saving the converted date in the database (col. 6, lines 1-3 et seq).

As to claim 10, Shaughnessy discloses the invention as discussed in the rejection of claim 9, as well as the claimed limitation of 'manipulating information in the database having the reformatted date information therein' by performing updates on the converted dates and saving said converted dates in the database (col. 6, lines 1-22 et seq).

The Examiner has also taken the position that:

As to claim 1, Ohms substantially discloses the invention including the claimed 'method of processing symbolic representations of dates stored in a database' by presenting a computer-implemented method for processing date outside the twentieth century (see title, p 244 et seq). In particular, Ohms discloses the step of 'providing a database with symbolic representations of dates stored therein according to a format wherein M.sub.1 M.sub.2 is the numerical month designator, D.sub.1 D.sub.2 is the numerical day designator, and - Y.sub.1 Y.sub.2 is the numerical year designator, all of the symbolic representations of dates falling within a 10-decade period of time' by detailing a short Gregorian format (MMDDYY) to represent dates (p 247, see table 1), wherein said dates fall within a 10 decade or a 100 year window (p 249, left-hand column, lines 3-7 et seq). Ohms also discloses the step of 'determining a century designator C.sub.1 C.sub.2 for each symbolic representation of a date in the database, Csub.1 C.sub.2 having 'a first value if Y.sub.1 Y.sub.2 is less than Y.sub.A Y.sub.B and having a second value if Y.sub.1 Y.sub.2 is equal to or greater than Y. sub. A Y. sub. B' by indicating that years that are later or equal (25-99) to the pivot date (25) would fall within the 20th century thereby equating C1 C2 to 19 (i.e. 1925-1999), whereas dates that are earlier (00-24) than the pivot date would fall within the 21st century thereby equating C1 C2 to 20 (i.e. 2000-2024) (see p 248, right-hand column. Finally, Ohms discloses the step of 'reformatting the symbolic representation of the date with the values C.sub.1 C.sub.2, Y.sub.1 Y.sub.2, M.sub.1 M.sub.2, and D.sub.1 D.sub.2 to facilitate further processing of the dates' by indicating that upon determining that a two-digit date falls within the 20th or the 21st the century[3], it is expressed in accordance with its corresponding century (i.e. 25-99 ---> 1925-1999, and 00-24 --->2000-2024) (p 248, right hand column).

[3] "Ohms implicitly discloses that CI C2 corresponds to 19 or 20 depending on whether the date is less than or greater than or equal to the pivot date.

Regarding the step of 'selecting a 10-decade window with a Y.sub.A Y.sub.B value for the first decade of the-window, Y.sub.A Y.sub.B being no later than the earliest Y.sub.1 Y.sub.2 year designator in the database,' Ohms discloses specifying a year as the desired starting point (pivot date) of the 100 year window (p 248, right hand column, 2nd paragraph). Ohms does not particularly detail that the pivot date is earlier than the earliest two digit date in the database. Hazama, however, discloses an analogous date processing system wherein, for a 100 year window, the pivot date for the window is selected based on a two digit year date that is smaller than the smallest two digit year date in the database (e.g. if smallest two digit date stored in database is 73), the pivot date for the 100 year window is chosen to be 72). (page 4 of translated document, last paragraph). It would have been obvious to one of ordinary skill in the art of data processing to combine the teachings of the cited references. The ordinary skilled artisan having read Ohms would immediately see the need to determine which 100 year span to use. This determination would have led the ordinary skilled artisan to the Hazama reference, which teaches the pivot date being smaller than the smallest two digit date in the database having all the dates within a 100 year period as a solution to restrict the selection of Ohms's window and thereby forcing all dates already stored in the database to fall in the 20th century.

As to claim 2, Ohms and Hazama disclose the invention as discussed in the rejection of claim 1. Additionally, Ohms discloses that 'the 10-decade window includes the decade beginning in the year 2000' by indicating that the 100 year window contains dates that span in the 21st century (2000-2024) (p 248, right hand column).

As to claim 3, Ohms and Hazama disclose the invention as discussed in the rejection of claim 2. Additionally, Ohms. discloses that the step of 'determining includes the step of determining the first value as 20 and the second value as 19' by indicating that dates that are greater or equal to the pivot date fall within the 20th century (C1 C2=19) and dates that are less than the pivot. date fall within the 21st century (C1 C2=20) p. 2488, right hand column).

As to claim 5, Ohms and Hazama disclose the invention as discussed in the rejection of claim 1. Additionally, Ohms discloses that the step of 'reformatting includes the step of reformatting each symbolic representation of a date into the format C.sub.1 C.sub.2 Y.sub.1 Y.sub.2 M.sub.1 M.sub.2 D.sub.1 D.sub.2 by indicating that dates that fall within the 20th century (greater than or equal to the pivot date) are preceded by 19 (e.g. 1925-1999), whereas dates that fall within the 21st century (less than the pivot date) are preceded by 20 (e.g. 2000-2024) p 2477, right hand column).

As to claim 7, Ohms and Hazama disclose the invention as discussed in the rejection of claim 1. Additionally, Ohms discloses that the step of 'providing a database includes the step of converting pre-existing date information having a different format into the format wherein M.sub.1 M.sub.2 is the numerical month designator, D.sub.1 D.sub.2 is the numerical day designator and Y.sub.1 Y.sub.2 is the numerical year designator' by suggesting the conversion from of a date from

> a Gregorian format (MMDDYYYY) to a short Gregorian format (YYMMDD), wherein YY indicates the year, MM indicates the month and DD indicates the date p 2477, table 1).
>
> As to claim 9, Ohms and Hazama disclose the invention as discussed in the rejection of claim 1. Additionally, Ohms discloses, 'after the step of reformatting, the storing of the symbolic representation of dates and their associated information back into the database' by suggesting that converted eight dates are stored in the database although they take up more memory space than non-converted six digits dates p 2499, left hand column).
>
> As to claim 10, Ohms and Hazama disclose the invention as discussed in the rejection of claim 9. Additionally, Ohms discloses, 'after the step of reformatting, the manipulating of information in the database having the reformatted date information therein' by suggesting that the converted dates can be saved in the database p 2499, left hand column).

United States Patent No. 5,630,118, issued to Shaughnessy on May 13, 1997, on an application filed on November 21, 1994, entitled, SYSTEM AND METHOD FOR MODIFYING AND OPERATING A COMPUTER SYSTEM TO PERFORM DATE OPERATIONS ON DATE FIELDS SPANNING CENTURIES ("Shaughnessy"), was before the Examiner during the original prosecution of the patent. The original Examiner was quite correct in not finding Shaughnessy applicable to the patentability of the claims as issued in the patent. Shaughnessy is not applicable to the claims added in the Reissue application either.

Shaughnessy was before the Examiner in the original prosecution of the Dickens patent being reexamined. The Examiner in the prosecution of the patent ("the original Examiner") clearly fully considered Shaughnessy and determined it not to be a reference under §102 or §103. The original Examiner was right, since Shaughnessy does not teach the claimed invention as recited in any of the claims of the patent, nor does it render the claimed invention obvious. In fact, like Ohms, it teaches away from the claimed invention.

Disclosed in Shaughnessy is a "system and method for *modifying and operating a computer system* to perform date operations on date fields having a two digit representation for the year without erroneously mistaking the years 2000 *et seq.* for the years 1900 *et seq.*" (Col. 1, lines 11-14)

As a solution to this problem, Shaughnessy proposes:

> [i]n accordance with the present invention, the current date operation routines nested in the body of the application program would be replaced with a call to one

> of a plurality of subroutines stored externally from the existing application program, *as opposed to the date operation routine being reprogrammed to perform the date operation in a new format.* The subroutines will be able to *accommodate the date format currently employed* by the application program, thus making it *unnecessary to convert all of the data fields in files containing data used in the application program over to the new data format.* (Col 4, lines 27-38).

As an example Shaughnessy describes a program that would "perform[] a date comparison to determine when loan payments are overdue … ." (Col.4, lines 39-40) According to the Shaughnessy method, the:

> program statements which performed the above functions would be modified to include program statements which did the following:
>
> 1. *Call the subroutine* which performs the date comparison *passing today's date, the date the next payment is due, and a three byte parameter*, the first byte of which identifies the format of today's date, the second byte of which identifies the format of the date next payment is due, and the third byte of which is left available for a *return code indicative of the result of the comparison*;
> 2. *If the result received from the subroutine indicates that the date next payment is due is greater than today's date, indicate that the account is okay.* (Col 4, lines 48-62)

In order to do this, Shaughnessy suggests that "for the subroutines to be able to accommodate different date formats, certain information, namely the current date, end of 100 year cycle, and two possible century values, must be determined and made available to the subroutines." (Col. 4, line 66 - Col. 5, line 3) In addition, Shaughnessy teaches that "*each subroutine that performs a date operation will include a call to another subroutine which can determine this information.*" (Col. 5, lines 3-5)

Further according to Shaughnessy "[t]he above-mentioned information will be used in the subroutine(s) to assign a century value to the two digit representation of the year of the dates to be operated on such that the subroutine can accurately perform its intended function." (Col. 5, lines 21-25) According to the Shaughnessy method this is performed using *a form of windowing* in which:

> [t]he current date is determined … in a format which utilizes a four digit representation for the year. Initially, the current date is set to the operating system date in the format 00YYDDD … by way of example … 0094263 … . *The current date is then compared to the date the system was installed* with the [date conversion] modifications (modified system install date) … which, for the sake of example, is 1994032 … . *If the YYDDD portion of the current date, 94263 is greater than or equal to the corresponding portion of this modified system install date, 94032 … then the century of the current date is set to the century of the modified system install date … .*" …
>
> If the current date appeared *less than the modified system install date* … in the 00YYDDD format … then the current date century would be *set in the format CCYYDDD to the century value for the modified system install date plus one* … .
>
> (Col 5, lines 31-57)

The Shaughnessy method then determines "the end of the 100 year cycle" according to "several parameters [which] may be specified." These "may include the number of years of future dating required (default is 10), … and whether the end of the 100 year cycle is to be updated daily … ." (Col. 6, lines 4-13) "If the cycle is to be updated daily, then the starting date is set to the current date …, as determined above. … Next, the end of the 100 year cycle is determined by adding the number of years future dating required to the starting date … ." (Col 6, lines 17-22).

Further, explains Shaughnessy:

> *The application program currently operating in a particular computer system may have a comparison of two date fields as part of its operation. If so, the source code which performs the comparison can be replaced* with a *call to DS2000R1, the name given to an exemplary comparison subroutine* useful in practicing the present invention … .
>
> …
>
> As illustrated in FIG. 8, the call DS2000R1 … is inserted into the application program, and includes parameters P1, P2, and P3. P1 and P2 are the *date fields which are to be compared.* For example, P1 could be "DATE-NEXT-PAYMENT-DUE" and P2 might be "TODAY" as referenced in the above sample

of modified source code. P3 is a three byte field in which the first two bytes define the type of date field P1 and P2, respectively. *The third byte is a return code which will be set to a value indicative of the result of the comparison.* (Col. 8, line 35 - Col. 9, line 53)

In summary, the teaching of Shaughnessy is to "[c]all the subroutine which performs the date [operation] passing [two dates] and the three byte parameter [including] a return code indicative of the result of the [operation]." Windowing occurs in the called subroutine in a manner other than that of the claimed invention. Moreover, this "on call" or "on the fly" windowing of at most two date data entries at a time is not the subject matter of the claimed invention.

There is, therefore, no teaching or suggestion in Shaughnessy of:

selecting a 10-decade window with a Y_A Y_B value for the first decade of the window, Y_A Y_B *being no later than the earliest Y_1 Y_2 year designator in the database*; A1

The Shaughnessy method selects a 10-decade window utilizing the "date the system was installed." 1

There is also, therefore, no teaching or suggestion in Shaughnessy of:

A2 *determining a century designator C_1 C_2 for each symbolic representation of a date in the database*, C_1 C_2 having ... ; .... .

The teaching of Shaughnessy is to determine a century designator for at most two date data representations being processed in a called subroutine at any given time.

There is also, therefore, no teaching or suggestion in Shaughnessy of:

A3 *reformatting the symbolic representation of the date with the values C_1 C_2, Y_1 Y_2, M_1 M_2, and D_1 D_2 to facilitate further processing of the dates.*

The teaching of Shaughnessy is to reformat two dates at a time in the called subroutine and the return to the program from the called subroutine of an indicator of the result of the processing of the two reformatted date data entries. Shaughnessy does not teach facilitating "further processing of the dates" by "reformatting the symbolic representation of the date" "for each symbolic representation of a date in the database."

Accordingly, there is also no teaching or suggestion in Shaughnessy of:

A4 *sorting the symbolic representations of dates*; (claim 4).

The method of Shaughnessy does not teach sorting all of the "symbolic representations of dates." It teaches only the comparison of two dates to each other in the called subroutine and returning to the program an indication of the result of the comparison.

There is also no teaching or suggestion in Shaughnessy of:

A5

reformatting each symbolic representation of a date into the format C_1 C_2 Y_1 Y_2 M_1 M_2 D_1 D_2 (claim 5), nor *sorting the symbolic representations of dates* using a numerical-order sort (claim 6); nor *storing the symbolic representation of dates* and their associated information back into the database (claim 9), nor manipulating information in the database *having the reformatted date information therein* (claim 10).

In addition, there is no teaching or suggestion in Shaughnessy of:

A6

converting pre-existing date information[within a database] having a different format into the format wherein M_1 M_2 is the numerical month designator, D_1 D_2 is the numerical day designator and Y_1 Y_2 is the numerical year designator (claim 7).

In addition, there is no teaching or suggestion in Shaughnessy of:

A7

selecting Y_A Y_B *such that Y_B is 0* (zero) (claim 8).

Claims 2 and 3 of the patent are dependent on claim 1 and are patentable along with claim 1. *In re Fine*, *supra.*

Japanese Published Patent Application, HEI 5-27947, entitled METHOD OF GUARANTEEING YEAR ORDER, with inventor Masakazu Hazama, published on February 5, 1993 ("Hazama") discloses a system "to guarantee the year order, even for years after 2000 AD, with the current file format, even when the year is managed by the last two digits of the date in digital files." (Hazama, at 2)

The system of Hazama, like Shaughnessy, discussed above, modifies a date in a record using a "correspondence utility module (10)." (Hazama, at 2) For that single record "the position in the record where the last two digits AD had been previously stored are specified" to the module 10 from "external parameter" 9. Hazama further notes that "the processing section will replace the code of the 10's place in the last two digits of the date with a code that maintains the year order." (Hazama, at 4)

A form of windowing is applied. Hazama notes that "the following processing is performed by the module (10)." (Hazama, at 5) This is then followed by "Work area output processing (5): data that have undergone replacement processing (4) are output to the work area (8)." (Hazama, at 6)

Referring to Figure 1 in Hazama, it is more clearly demonstrated that Hazama is not more applicable to the patentability of the claimed invention than Shaughnessy, as discussed above. The data from a record is moved from the processor "work area" 8 to the "Year 2000 date correspondence utility module" 10. A modified date, after some form of windowing, for that single record is returned to the work area 8 for processing. Hazama, like Shaughnessy, therefore, does not disclose or suggest the claimed invention.

There is also, therefore, no teaching or suggestion in Hazama nor in the combination of Shaughnessy and Hazama of:

B1 *determining a century designator C_1 C_2 for each symbolic representation of a date in the database*, C_1 C_2 having … ; … .

The teaching of Hazama, or Shaughnessy in view of Hazama, is to determine a century designator for at most two date data representations being processed in a called subroutine/module at any given time.

B2 There is also, therefore, no teaching or suggestion in Hazama, or Shaughnessy in view of Hazama, of:

reformatting the symbolic representation of the date with the values C_1 C_2, Y_1 Y_2, M_1 M_2, and D_1 D_2 to facilitate further processing of the dates.

The teaching of Shaughnessy, or Shaughnessy in view of Hazama, is to reformat at most two dates at a time in the called subroutine/module and the return to the program from the called subroutine/module of an indicator of the result of the processing of the two reformatted date data entries or a single modified date from a single record. Neither Shaughnessy nor Shaughnessy in view of Hazama teaches facilitating "further processing of *the dates*" by "*reformatting* the symbolic representation of *the date*" "*for each symbolic representation of a date in the database*." B3

Accordingly, there is also no teaching or suggestion in Shaughnessy, or in Shaughnessy in view of Hazama, of:

sorting the symbolic representations of dates; (claim 4).

B4

The method of Shaughnessy, or of Shaughnessy in view of Hazama, does not teach sorting all of the "symbolic representations of dates." It teaches only the comparison of two dates to each other in the called subroutine or changing a single date representation from a single record and returning to the program an indication of the result of the comparison or the changed single date record.

There is also no teaching or suggestion in Shaughnessy, or Shaughnessy in view of Hazama, of:

B5

reformatting each symbolic representation of a date into the format C_1 C_2 Y_1 Y_2 M_1 M_2 D_1 D_2 (claim 5), nor *sorting the symbolic representations of dates* using a numerical-order sort (claim 6); nor *storing the symbolic representation of dates* and their associated information back into the database (claim 9), nor manipulating information in the database *having the reformatted date information therein* (claim 10).

In addition, there is no teaching or suggestion in Shaughnessy, or in Shaughnessy in View of Hazama, of:

B6

converting pre-existing date information[within a database] having a different format into the format wherein M_1 M_2 is the numerical month designator, D_1 D_2 is the numerical day designator and Y_1 Y_2 is the numerical year designator (claim 7).

In addition, there is no teaching or suggestion in Shaughnessy, or in Shaughnessy in view of Hazama, of:

B7

selecting Y_A Y_B *such that Y_B is 0* (zero) (claim 8).

Claims 2 and 3 of the patent are dependent on claim 1 and are patentable along with claim 1. *In re Fine, supra.*

Clipper 5A, described in Booth, in a fashion very similar to Ohms' use of the Lilian date format, as discussed below, operates with date data stored in databases representing dates in the form of a number, often referred to as integer date formatting. Each unique number represents a date or a time and date down to a specific time increment, e.g., a mili-second, starting with a certain date or date/time and counting to the limits of the number of binary places available, e.g., 32, incrementing the chosen time increments.[4] As noted, each integer could represent a day or day/time incremented by the value of the integer from a start date or day/time. That is to say, the total time period that can be represented depends upon the starting date or date/time, the number of unique combinations of, e.g., 32 bits, and the increment counted, e.g., days, seconds, mili-seconds, etc.[5]

It is also true of dates stored in integer format, as is also the case with the dates stored in Lilian format, that *there is no Y2K ambiguity problem in regard to dates stored in a database in these formats. Booth, therefore, like Ohms, does not even suffer from the problem that the claimed invention is meant to address. Knowing the starting point (the starting date in Lilian, or the starting date or date/time in integer), the granularity (i.e., one day in Lilian and, e.g., one day or one second or one mili-second in integer), and the incremental difference between the starting point and the integer date number stored in either the Lilian or integer format, the stored date is known, including the year to four digits, and, therefore, including within what is so stored the information needed to determine a century designator. No possible Y2K ambiguity problem can exist or does exist when storing dates in a database in these formats. The claimed invention does not relate to databases with dates stored in these formats. The claimed invention involves databases with dates stored in them where there is an ambiguity because only two digits of date data are present in the stored information, from which to determine the full date, including a century designator. Every date stored in a database in Lilian or integer* *

is this true?

[4] This is the same as the Lilian format disclosed in Ohms, however, only the date is stored in Lilian format, according to the teaching of Ohms, and the Lilian date format, strictly speaking, starts with the beginning of the Gregorian calendar. Ohms, however, does disclose a modified Lilian format starting at some other arbitrarily selected starting date. According to the discussion in Booth, at 949-951, Clipper may count integers for year/date separately from hours, minutes and seconds on a given date, but Booth discloses at 951 using a single number, an integer for the year/date, and a decimal component for the elapsed seconds from midnight on the particular year/date. This simply makes Clipper even more identical to Ohms' disclosure of a modified Lilian integer date with the integer representing some day incrementally counted from some starting day, spanning over several centuries. In this format, as noted, each incremental number represents a day or a moment in time, *each such day or moment being, by definition, a part of a fully defined and recoverable four digit year date datum.*

[5] Booth, at 939 ("Dates are stored internally in such a way that math operations can be performed on dates to derive other dates. Adding an integer to a date will result in a future date. Subtracting two dates will result in a number of days between the two.") Booth, at 99 ("The date type is used to represent calendar dates. Clipper stores dates internally in such a way that a variety of operations can be performed on them. You can determine the number of days between two dates by subtracting them, and you can determine a future date by adding an integer value to a date value. The result will be a date value, some number of days in the future.")

format, by definition, already has a century designator; can not possibly be ambiguous due to reaching the end of a century[6]*; and never needs to have a century designator determined, whether by the method of the claimed invention or otherwise.*

For this reason alone, Booth, like Ohms, has nothing to do with the patentability of the claimed invention, and certainly does not anticipate the claimed invention. In fact, it teaches away from the claimed invention.

Booth describes a number of functions that the programming language utilizes to read or write dates into the database, display dates on a screen, find the difference between two dates or a date/time equal to a given date/time plus some incremental time period, to find the day of the week of a given date/time, and like functions. Like Ohms and Shaughnessy, some of these functions employ windowing in some fashion or another. Like Shaughnessy, when they do employ windowing these programming routines "[c]all the subroutine which performs the date [operation] passing [information] indicative of the result of the [operation]." Windowing occurs in the called subroutine in a manner other than that of the claimed invention. Such "on call" or "on the fly" windowing of at most two date data entries at a time is not the subject matter of the claimed invention.

By way of example, Booth includes "an easy way to validate a character string used as a date," and/or check for "correctly formatted dates that are not reasonable or even possible," and/or to select a "'safety' date which no [date being entered] can precede [, or] maximum allowable date." (Booth, at 526-28) In addition Booth describes "date manipulation capabilities." Clipper "provides three settings which control the display of dates," [e.g.,] "whether or not the year portion of a date is display [sic] with four digits (including the century) or two digits (not including the century)," [and] "different date display formats." (*Id.* at 939-40)

Booth also describes a "SET EPOCH command" which, in the same way as Ohms, "informs the system how to handle *date data entry* that use only two digits for the year." According to this function, "[w]hen a two-digit year is entered into a date, its year digits are compared with the year digits of the epoch setting to determine the century to

[6] Even at the end of the time span that can be represented, e.g., as noted below Clipper dates run until December 31, 2999, there is no "ambiguity." The system simply cannot express a date beyond the given

place the date into. If the two digits are prior to the setting of SET EPOCH, the year is assumed to be in the next century. If the digits are greater than or equal to the SET EPOCH setting, the year is assumed to be in the current century." (*Id.*, at 941)

This is *virtually identical to the utilization of windowing to enter dates into a database (where they are then stored in Lilian format) that is disclosed in the Ohms article*, as discussed by the applicant as noted below. In the SET EPOCH function disclosed in Booth, the pivot year defaults to 1900, which "forces any date entered to be considered a date in the twentieth century." This also means that in this mode, no Y2K date ambiguity problem is recognized or accommodated.

Booth also discloses a function DTOC(), i.e., "[t]he date to character function [which] takes a date variable ... and returns a string representation of the date. The string is recreated in the format specified by the SET DATE or the SET DATEFORMAT command. If SET DATE has not been specified, the default date format is mm/dd/yy." (*Id.*, at 944) Also disclosed is a function DTOS() "date to string function" which "takes a date variable ... and returns a string in the format YYYYMMDD ... ." The formats available for SET DATE are set forth on page 940 of Booth.

There is, therefore, no teaching or suggestion in Booth of:

> providing a database with *symbolic representations of dates stored therein according to a format wherein* M_1M_2 is the numerical month designator, D_1D_2 is the numerical day designator, and Y_1Y_2 is the numerical year designator, a*ll of the symbolic representations of dates falling within a 10-decade period of time*."

Booth's "method of processing symbolic representations of dates stored in a database" teaches utilizing a database with the symbolic representations of dates stored therein in the form of unique integers or numbers, each representative of a unique day or other more granular moment in time, or a combination of a unique day and a number representing a unique time on that day. This is not storage in a M_1M_2, D_1D_2, Y_1Y_2 format. In addition, there is no teaching or suggestion that those dates all fall within a 10-decade period of time. So far as Booth teaches the dates stored in the database can be

span (without changing the parameters, e.g., the length of the integer number or the granularity).

any span of dates capable of being represented over the span of time capable of being represented by the particular integer date system being used.[7]

There is also no teaching or suggestion in Booth of:

selecting a 10-decade window with a Y_A Y_B value for the first decade of the window, Y_A Y_B *being no later than the earliest Y_1 Y_2 year designator in the database*; … .

Booth selects, e.g., "nyear" in order to "handle dates that use only two digits for the year [w]hen a two-digit year is *entered into a date* [by comparing] its year digits … with the year digits of the epoch setting to determine the century … ," (*Id.*, at 941). There is no teaching or suggestion of any consideration of "*the earliest Y_1 Y_2 year designator in the database.*"

There is also, therefore, no teaching or suggestion in Booth of:

determining a century designator C_1 C_2 for each symbolic representation of a date in the database, C_1 C_2 having … ;

There is no need to determine a century designator for each symbolic representation of a date in Booth's database, since each is already stored with the century designator included in the date datum so stored in integer format. In addition, the teaching of Booth is to determine a century designator on an individual date datum basis for date data entry, date display, incrementally determining a date based upon a given initial date datum, etc. This calling of certain functions disclosed by Booth to, for example, display a date, or compare two dates, or increment a date from a starting date, are virtually identical to the pertinent disclosure in Shaughnessy, discussed by the patent owner as noted above. As has been noted by the patent owner before, Shaughnessy was properly considered by the original Examiner not to have been relevant to the patentability of the claimed invention.

There is also, therefore, no teaching or suggestion in Booth of:

reformatting the symbolic representation of the date with the values C_1 C_2, Y_1 Y_2, M_1 M_2, and D_1 D_2 to facilitate further processing of the dates.

Booth, like Ohms, does not need to do the recited reformatting, since the dates stored in the database in their original format already contain all the information needed

[7] Booth, at 99 ("Clipper supports all dates from January 1, 100AD through December 31, 2999.")

to determine the four digit designation of the date, including the century of the particular date datum. The process of the claimed invention is not needed for dates stored with the century designator already known from what is stored and the Y2K ambiguity not present. Furthermore, the teaching of Booth, like Shaughnessy, is to reformat one or two dates at a time in a called Clipper date functionality and the return to the program from the called subroutine with information resulting from the performance of the programming functionality, e.g., an input to a display, a result of a comparison, a newly calculated date, etc. Booth does not teach facilitating "further processing of the dates" by "reformatting the symbolic representation of the date" "for each symbolic representation of a date in the database."

Accordingly, there is also no teaching or suggestion in Booth of "*sorting the symbolic representations of dates*," as recited in claim 4. These are the reformatted symbolic representations. *Whatever sorting Booth teaches does not need to first reformat the date data, since the integer format can be and is sorted in its initial format. The method of the claimed invention, including the reformatting steps is simply not relevant to a database that stores date data as Clipper does, in integer format, as described in Booth.*

There is also no teaching or suggestion in Booth of "*reformatting each symbolic representation of a date into the format $C_1 C_2 Y_1 Y_2 M_1 M_2 D_1 D_2$*," as recited in claim 5. Neither is there a disclosure of "*sorting the symbolic representations of dates* using a numerical-order sort," as recited in claim 6. There is no disclosure of "*storing the symbolic representation of dates* and their associated information back into the database," as recited in claim 9 nor "manipulating information in the database *having the reformatted date information therein*," as recited in claim 10.

In addition, there is no teaching or suggestion in Booth of "*converting pre-existing date information [within a database] having a different format into the format* wherein M_1 M_2 is the numerical month designator, D_1 D_2 is the numerical day designator and Y_1 Y_2 is the numerical year designator," as recited in claim 7. This process step is recited as part of the "step of providing the database" upon which the subsequent process steps recited in the claimed invention, are carried out. The fact that Booth, or other art, teaches converting date data from one format into the recited format, does not teach it as part of the process of the claimed invention. Similarly, there is no teaching or suggestion in Booth of "selecting $Y_A Y_B$ *such that Y_B is 0* (zero)," as recited in claim 8, even though SET EPOCH can and does use pivot years ending in 0. SET EPOCH, as noted above, is not a process according to the claimed invention.

Claims 2 and 3 of the patent are dependent on claim 1 and are patentable along with claim 1.

Ohms teaches a "[m]ethod[] of *using existing date formats across century boundaries* .... *The use of a format termed the Lilian date format ... is introduced.*" (Ohms, at 244, Abstract) Ohms teaches that "[t]he two positions traditionally used in both Julian and Gregorian date formats implicitly represent a year within a century. However, this system is inadequate for representing dates in more than one century." (*Id.* at 245) As a solution Ohms proposes a "Lilian date format [to] avoid[] the ambiguity by *using seven positions* for the *number of the days from the beginning of the Gregorian calendar, October 15, 1582.*" (*Id.* at 245) "The value is incremented by one for each subsequent day." (*Id.* at 246) Ohms explains that "the *Lilian date format* is presented here as the *basis for making date conversions* .... This format *handles processing across century years and other aspects of date conversion not currently adaptable to computer programming.*" (*Id.* at 244-45)

In this context, of database conversion to Lilian format from more traditional Gregorian or Julian formats, Ohms describes under the heading "Accommodating end users" the fact that they "*usually enter two digits for the year in a date and understand the ambiguity that this represents.*" (*Id.* at 248) Ohms goes on to say that:

> *to avoid adverse user reaction, [by requiring the entry of date data in other than two digits] programs must continue to function with only two digits for year. The inference of the year 1997 from 97 and 2003 from 03 must continue. For the exceptional case where the correct meaning could be 1897 and 1903, entry of all four digits may be required.* (*Id.* at 248)

It is in this context that Ohm notes:

> it may be necessary to provide a *conversion function* that receives a definition of the implied century as a parameter. An excellent way to do this unambiguously is to specify a year as the desired starting point of a 100-year range. For example, if

the starting year for the range is specified as 1925, dates with year digits between 25 and 99 would be between 1925 and 1999, and dates with year digits of 00 through 24 would lie between 2000 and 2024. (*Id.* at 248)

Ohms therefore, simply teaches storing dates in a database in Lilian format which "handles processing across century years" and "[a]ccommodating end users" who "enter two digits for the year" by "providing a conversion function" using windowing for data entry. Simply because the claimed invention employs a known technique, i.e., windowing, does not, of itself, make the process of the claimed invention obvious. Hawes §7.05, at 7-8.1, *citing*, *In re Brower*, 77 F.3d 422, 37 U.S.P.Q.2d 1663 (Fed. Cir. 1994).

There is, therefore, no teaching or suggestion in Ohms of:

providing a database with symbolic representations of dates stored therein according to a format wherein M_1 M_2 is the numerical month designator, D_1 D_2 is the numerical day designator, and Y_1 Y_2 is the numerical year designator, all of the symbolic representations of dates falling within a 10-decade period of time; … .

Unlike this recitation of claim 1, Ohms teaches providing a database with the dates in a Lilian format.

There is also, therefore, no teaching or suggestion in Ohms of:

providing a database … *all of the symbolic representations of dates falling within a 10-decade period of time*; … .

Ohms teaches having data in the database in Lilian format, i.e., within a ninety-nine million day window (seven chronological day date numbers starting at a given date).

There is also, therefore, no teaching or suggestion in Ohms of:

selecting a 10-decade window with a Y_A Y_B value for the first decade of the window, Y_A Y_B *being no later than the earliest Y_1 Y_2 year designator in the database*; … .

At best Ohms teaches or suggests selecting a Y_AY_B based upon dates that are currently being input into the database.

There is also, therefore, no teaching or suggestion in Ohms of:

determining a century designator $C_1\ C_2$ for each symbolic representation of a date in the database, $C_1\ C_2$ having ... ; ... .

Ohms teaches entering date data into the database to be converted into Lilian format for storage and manipulation within the database. He does not teach or suggest determining a century designator for data in the database. Lilian format needs none.

There is also, therefore, no teaching in Ohms of:

reformatting the symbolic representation of the date with the values $C_1\ C_2$, $Y_1\ Y_2$, $M_1\ M_2$, and $D_1\ D_2$ to facilitate further processing of the dates.

Ohms teaches reformatting into Lilian format and thereafter processing the date data in the database utilizing the Lilian format.

Therefore, Ohms does not teach the claimed invention as recited in claim 1 for purposes of 35 U.S.C. §102. In addition Ohms would not have made the claimed invention as recited in claim 1 obvious to a person of ordinary skill in the art at the time the invention was made, under 35 U.S.C. §103. As explained above, not only does Ohms not teach or suggest the claimed invention recited in claim 1, it clearly teaches away from virtually every step of the method of the claimed invention as recited in claim 1.

For like reasons, there is also no disclosure or suggestion in Ohms of:

sorting the symbolic representations of dates (claim 4); or *reformatting each symbolic representation of a date into the format $C_1\ C_2\ Y_1\ Y_2\ M_1\ M_2\ D_1\ D_2$* (claim 5); or s*orting the symbolic representations of dates* using a numerical-order sort (claim 6); or *storing the symbolic representation of dates* and their associated information back into the database (claim 9).

There is also no teaching or suggestion in Ohms of:

The method of claim 9, including the additional step, after the step of reformatting, of

manipulating information in the database *having the reformatted date information therein* (claim 10).

In addition, there is no teaching or suggestion in Ohms of:

converting pre-existing date information having a different format into the format wherein $M_1\ M_2$ is the numerical month designator, $D_1\ D_2$ is the numerical day designator and Y_1 Y_2 is the numerical year designator (claim 7) or selecting Y_A Y_B *such that Y_B is 0* (zero) (claim 8).

Claims 2 and 3 of the patent are dependent on claim 1 and are patentable along with claim 1. *See*, *In re Fine*, 837 F.2d 1071, 5 U.S.P.Q.2d 1596 (Fed. Cir. 1988).

Claims 4, 6, 8 stand rejected under 35 U.S.C.§103(a) as being unpatentable over Shaughnessy in view of Booth or, in the alternative, under 35 U.S.C.§103(a) as obvious over Shaughnessy in view of Hazama, as applied to the rejection of claims 1-3 5, 7, 9-10 above, further in view of Booth.

Claims 4, 6, 8 stand rejected under 35 U.S.C.§103(a) as being unpatentable over Ohms in view of Hazama, as applied to the rejection of claims 1-3, 5, 7, 9-10 above, further in view of Booth.

The Examiner has taken the position that:

> As to claim 4, Shaughnessy substantially discloses the invention as discussed in the rejection of claim I above. Shaughnessy does not, disclose, the additional step of 'sorting the symbolic representations of dates, after the step of reformatting.' However., Booth discloses an analogous system that utilizes the Clipper programming language to process dates stored in a database to thereby derive other dates therefrom (p. 939, lines 1-3 et seq. In particular, analogously to Shaughnessy, Booth discloses the SET EPOCH command for comparing an entered two digit date with the year digit (pivot date) of the epoch setting to determine the century to place the date into (p. 94 1, see SET EPOCH paragraph et seq) thereby converting a six digit date (MMDDYY) into a. corresponding eight. digit date (C1C2Y1Y2M1M2D1D2) (see p. 940-941). Additionally., Booth complements Shaughnessy by suggesting the sorting of converted dates after having been reformatted by the SET EPOCH command (p. 945). It would have been obvious to one of ordinary skill in the art of data processing at the time of the instant invention to combine the teachings of the cited references. Booth's teaching of sorted reformatted dates would facilitate Shaughnessy's system to return the reformatted dates in chronological sequence. And it would therefore be very useful when indexing the database in date order, as suggested by Booth in page 945.
>
> As to claim 6, Shaughnessy substantially discloses the invention as discussed in the rejection of claim 5 above. Shaughnessy does not specifically, disclose the additional step of 'sorting the symbolic representations of dates using a numerical-order sort, after the step of reformatting.' However, Booth discloses an analogous system that utilizes the Clipper programming language to process dates stored in a database to thereby derive other dates therefrom (p. 939, lines 1-3 et seq. In particular, analogously to Shaughnessy, Booth discloses the SET EPOCH command for comparing an entered two digit date with the year digit (pivot date), of the epoch setting to determine the century to place the date into (p. 941, see

SET. EPOCH paragraph et. seq) thereby converting a six digit date (MMDDYY) into a corresponding eight digit date (C1C2Y1Y2M1M2D1D2) (see p. 940-941). Additionally, Booth complements Shaughnessy by suggesting the sorting of converted dates after having been reformatted by the SET EPOCH command (p 945). It would have been obvious to one of ordinary skill in the art of data processing at the time of the instant invention to combine the teachings of the cited references. Booth's teaching of sorted reformatted dates would facilitate Shaughnessy's system to return the reformatted dates in chronological sequence. And it would therefore be very useful when indexing the database in date order, as suggested by Booth in page 945.

---------Elaborate Above--------------

As to claim 8, Shaughnessy substantially discloses the invention as discussed in the rejection of claim 1. Shaughnessy does not specifically, disclose the step of selecting Y.sub.A Y.sub.B such that Y.sub.B is 0 (zero). However, Booth discloses an analogous system that utilizes the Clipper programming language to process dates stored in a database to thereby derive other dates therefrom (p. 939, lines 1-3 et seq. In particular, analogously to Shaughnessy, Booth discloses the SET EPOCH command for comparing an entered two digit date with the year digit (pivot date) of the epoch setting to determine the century to place the date into (p. 941, see SET EPOCH paragraph et seq) thereby converting a six digit date (MMDDYY) into a corresponding eight digit date (C1C2Y1Y2M1M2D1D2) (see p. 940-941). Additionally, Booth complements Shaughnessy by suggesting that the pivot date be set to 90 by selecting set epoch to be 1990 (i.e. YAYB = 90), such that YB equals to zero (p. 942). It would have been obvious to one of ordinary skill in the art of data processing at the time of the instant invention to combine the teachings of the cited references. Booth's teaching of setting the pivot date to a predetermined value would enable users of Shaughnessy's system to have direct control over the return of reformatted dates, to thereby preset the date processing system in accordance with their needs.

The Examiner has also taken the position that:

As to claim 4, Ohms and Hazama substantially disclose the invention as discussed in the rejection, of claim 1 above. Ohms and Hazama do not particularly disclose the additional step of 'sorting the symbolic representations of dates, after the step of reformatting.' However,, Booth discloses an analogous system that utilizes the Clipper programming language to process dates stored in a database to thereby derive other dates therefrom p 939, lines 1-3 et seq. In particular, analogously to Ohms, Booth discloses the SET EPOCH command for comparing an entered two digit date with the year digit (pivot date) of the epoch setting to determine the century to place the date into p 9411, see SET EPOCH paragraph et seq) thereby converting a six digit date (MMDDYY) into a corresponding eight digit date (CIC2YIY2MIM2DID2) (see p. 940-941). Additionally, Booth complements Ohms and Hazama by suggesting the sorting of converted dates after having been reformatted by the SET EPOCH command (p. 945). It would have been obvious to one of ordinary skill in the art of data processing at the time of the instant invention to combine the teachings of the cited references. Booth's teaching of sorted reformatted dates would facilitate the Ohms-Hazama's system to return the reformatted dates in chronological sequence. And, it would therefore be very useful when indexing the database in date order, as suggested by Booth in page 945.

As to claim 6, Ohms and Hazama substantially disclose the invention as discussed in the rejection of claim 5 above. Ohms and Hazama do not specifically, disclose the additional step of 'sorting the symbolic representations of dates using a numerical-order sort, after the step of reformatting.' However Booth discloses an analogous system that utilizes the Clipper programming language to process dates stored in a database to thereby derive other dates therefrom (p 939, lines 1-3 et seq. In particular, analogously to Ohms, Booth discloses the SET EPOCH command for comparing an entered two digit date with the year digit (pivot date) of the epoch setting to determine the century to place the date into (p 94 1, see SET EPOCH paragraph et seq) thereby converting a six digit date (MMDDYY) into a corresponding eight digit date (C I C2Y I Y2M I M21) I D2) (see p. 940-94 1). Additionally, Booth complements Ohms and Hazama by suggesting the sorting of converted dates after having been reformatted by the SET EPOCH command (p. 945). It would have been obvious to one of ordinary skill in the art of data processing at the time of the instant invention to combine the teachings of the cited references. Booth's teaching of 'sorted reformatted dates would facilitate the Ohms-Hazama's system to return the reformatted dates in chronological sequence. And, it would therefore be very useful when indexing the database in date order, as suggested by Booth in page 945.

As to claim 8, Ohms and Hazama substantially disclose the invention as discussed in the rejection of claim 1. Ohms. and Hazama do not specifically, disclose the step of selecting Y.sub.A Y.sub.B such that Y.sub.B is 0 (zero). However, Booth discloses an analogous system that utilizes the Clipper programming language to process dates stored in a database to thereby derive other dates therefrom (p 939, lines 1-3 et-seq. In particular, analogously to Ohms and Hazama, Booth discloses the SET EPOCH command for comparing an entered two digit date with the year digit (pivot date) of the epoch setting to determine the century to place the date into (p 941, see SET EPOCH paragraph et seq) thereby converting a six digit date (MMDDYY) into a corresponding eight digit date (C1C2Y1Y2M1M2D1D2) (see p. 940-941). Additionally, Booth complements Ohms and Hazama by suggesting that the pivot date be set to 90 by selecting set epoch to be 1990 (i.e. YAYB=90), such that YB equals to zero (p. 942). It would have been obvious to one of ordinary skill in the art of data processing at the time of the instant invention to combine the teachings of the cited references. Booth's teaching of setting the pivot date to a predetermined value would enable users of the Ohms-Hazama's system to have direct control over the return of reformatted dates to thereby preset the date processing system in accordance with their needs.

Claims 11-15 stand rejected under 35 U.S.C.§103(a) as being unpatentable over Shaughnessy in view of Booth, or in the alternative over Shaughnessy in view of Hazama, and further in view of Booth.

Claims 11-15 stand rejected under 3 5 U. S. C.§103 (a) as being unpatentable over Ohms in view of Hazama, further in view of Booth.

The Examiner has applied the references as above with respect to the corresponding claims in claims 1-10. Similarly, to the discussion above, there is no teaching or suggestion in Shaughnessy, Hazama, Ohms or Booth, or any combination of these references of the claimed invention as recited in any of the claims 11-15 of the patent.

Claims 16-18, 20, 22, 24-25 stand rejected under 35 U.S.C.§102(e) as anticipated by Shaughnessy or, in the alternative, under 35 U. S.C.§103 (a) as obvious over Shaughnessy in view Hazama.

Claims 16-18, 20, 22, 24-25 stand rejected under 35 U. S.C. 103(a) as being unpatentable over Ohms in view of Hazama.

The Examiner has applied the references as above with respect to the corresponding claims 1-3, 5, 7, 9, and 11-12.

Claims.19, 21, 23 stand rejected under 35 U.S.C.§103(a) as being unpatentable over. Shaughnessy in view of Booth or, in the alternative, under 35 U.S.C.§103(a) as -----
----

obvious over Shaughnessy in view of Hazama, as applied to the rejection of claims 16-18, 20, 22, 24-25 above, further in view of Booth.

Claims 19, 21, 23 stand rejected under 35 U.S.C.§103(a) as being unpatentable over Ohms in view of Hazama, as applied to the rejection of claims 16-18, 20, 22, 24-25 above, further in view of Booth.

The Examiner has applied the references as above with respect to the corresponding claims 4, 6 and 8.

With respect to claim 16, neither Shaughnessy, nor Ohm, nor Booth nor Hazawa, nor any combination of these references discloses or suggests the <u>claimed invention</u> as recited in Claim 16. At a minimum, these references separately or collectively fail to teach or suggest "reformatting the symbolic representation of each symbolic representation of a date in the database … ." Further, they do not teach of ------(or) -------
---

suggest doing the reformatting “without the addition of any new data field to the database … .” In addition there is no teaching or suggestion of “the reformatted representation of each date in the database having the values C_1 C_2, … , in order to facilitate collectively further processing the reformatted symbolic representations … of each of the dates.”

Ohms and Booth, utilizing Lillian and integer date formats, respectively, do not reformat dates in the database at all, and do not even have the Y2K ambiguity problem addressed by the present invention. Shaughnessy and Hazama encounter a date datum and call a subroutine or module to process the single date, or at most two dates, for resolution of the Y2K ambiguity problem. This is not the claimed invention, as distinguished from these references by at least the recitations noted above.

As to claim 17, the claim depends from allowable claim 16, and should be allowed for that reason. *In re Fine, supra.*

As to claim 18, the claim depends indirectly from allowable claim 16, and shuld ------------(should)-------be allowed for that reason. *In re Fine, supra.*

As to claim 19, neither Shaughnessy, nor Ohm, nor Booth nor Hazawa, nor any combination of these references discloses or suggests the claimed invention as recited in Claim 19. For the reasons noted above, the “symbolic representations of [each of the] dates [in the database]” is not taught to be produced, and/or is not taught to be produced according to the method of the claimed invention, as recited in claim 16. Therefore, the “sorting” recited in claim 19 is also not as recited in the claimed invention.

The same is true with respect to claim 20, as was the case with claim 19, as noted above. In addition the references do not teach or suggest “reformatting each symbolic representation of a date … separately from the symbolic representations in the database.”

With respect to claim 21, the claim depends from an allowable claim 20, and should be allowed for that reason. *In re Fine, supra.*

With respect to claim 22, to the extent that the references or any of them discloses or suggests “converting pre-existing date information … ” as recited in claim 22, they do not teach or suggest doing so as a part of the method recited in the allowable claim 16.

With respect to claim 23, the same can be said as with respect to claim 22 above.

With respect to claim 24, neither Ohms, Shaughnessy, Booth nor Hazama, nor any combination of these references teach or suggest “storing the [reformatted] symbolic representation of dates and their associated information back into the database." Ohms and Booth do not store reformatted symbolic representations of dates, as recited, but instead store dates in, respectively, Lilian or integer date formats. Shaughnessy and Hazama do not send any reformatted date data for storage anywhere outside at most the subroutine/module called to handle one or two date representations at any given execution of the subroutine/module.

With respect to claim 25, neither Ohms, Shaughnessy, Booth nor Hazama, nor any combination of these references, teach or suggest "manipulating information in the database having the reformatted date information therein." Ohms and Booth do not manipulate reformatted symbolic representations of dates, as recited, but instead manipulate dates in, respectively, Lilian or integer date formats. Shaughnessy and Hazama do not manipulate any reformatted date information in the database.

Claims 26-30 stand rejected under 35 U.S.C.§103(a) as being unpatentable over Shaughnessy in view of Booth, or in the alternative over Shaughnessy in view of Hazama, and further in view of Booth.

Claims 26-30 are rejected under 35 U.S.C. 103(a) as being unpatentable over Ohms in view of Hazama, further in view of Booth.

With respect to claims 26-30, for the same reasons as just discussed with respect to the corresponding claims in claims 16-25, claims 26-30 should be allowed.

Claims 31-33 stand rejected under 35 U.S.C. 102(e) as anticipated by Shaughnessy, or in the alternative, under 35 U.S.C. 103(a) as obvious over Shaughnessy in view of Hazama.

Claims 31-33 stand rejected under 35 U.S.C.§103(a) as being unpatentable, over Ohms, in view of Hazama.

The Examiner has applied the references as above.

With respect to claim 31, neither Shaughnessy, nor Ohm, nor Booth nor Hazama, nor any combination of these references discloses or suggests the claimed invention as recited in Claim 31. At a minimum, these references separately or collectively fail to teach or suggest "reformatting the symbolic representation of each symbolic representation of a date in the database … ." Further, they do not teach or suggest doing the reformatting "without the addition of any new data field to the database … ." In addition there is no teaching or suggestion of "the reformatted representation of each date in the database having the values C_1 C_2, ... , in order to facilitate collectively further processing the reformatted symbolic representations ... of each of the dates."

Ohms and Booth, utilizing Lillian and integer date formats, respectively, do not reformat dates in the database at all, and do not even have the Y2K ambiguity problem addressed by the claimed invention. Shaughnessy and Hazama encounter a date datum and call a subroutine or module to process the single date, or at most two dates at any given execution of the subroutine/module for resolution of the Y2K ambiguity problem. This is not the claimed invention, as distinguished from these references by at least the recitations noted above.

With respect to claim 32, neither Shaughnessy, nor Ohm, nor Booth nor Hazawa, nor any combination of these references, discloses or suggests the claimed invention as recited in Claim 32. At a minimum, these references separately or collectively fail to teach or suggest "reformatting the symbolic representation of each of the dates in the database ... ." Further, they do not teach or suggest doing the reformatting "without the addition of any new data field to the database ... ." In addition there is no teaching or suggestion of "the reformatted representation of each date in the database having the values C_1 C_2, ... , in order to facilitate collectively further processing the reformatted symbolic representations ... of each of the dates."

Ohms and Booth, utilizing Lillian and integer date formats, respectively, do not reformat dates in the database at all, and do not even have the Y2K ambiguity problem addressed by the claimed invention. Shaughnessy and Hazama encounter a date datum and call a subroutine or module to process the single date, or at most two dates, in any given execution of the subroutine/module, for resolution of the Y2K ambiguity problem. This is not the claimed invention, as distinguished from these references by at least the recitations noted above.

With respect to claim 33, the same arguments apply to claim 33 as do to claim 32 as noted above.

Claims 34-59 stand rejected under 35 U.S.C. 103(a) as being unpatentable over Shaughnessy in view of Booth, or in the alternative, over Shaughnessy in view of Hazama, and further in view of Booth.

Claims 34-59 stand rejected under 35 U.S.C.§103(a) as being unpatentable over Ohm's in view of Hazama, further in view of Booth.

The Examiner has applied the references as above. In addition the Examiner has taken the position that:

> As to claim 35, Shaughnessy and Booth disclose the invention as discussed in the rejection of claim 34. Additionally, Shaughnessy discloses step of 'opening the database prior to the step of converting' by providing a subroutine to retrieve a six digit date from its storage location in an existing application program (i.e. requires opening the DB, first) prior to converting said date to an eight digit, format (col. 4, lines 29-31-et seq).
> As to claim 35, Ohms, Hazama and Booth disclose the invention as discussed in the rejection of claim 34. Additionally, Ohms discloses step of 'opening the database prior to the step of converting' by providing a subroutine to retrieve a six digit date from its storage location in an existing application program (i.e. requires opening the DB, first) prior to converting said date to an eight digit format (p 248, right hand column et seq).
> As to claims36-39, Shaughnessy and Booth disclose the invention as discussed in the rejection of claim 35. Booth 9 further complements Shaughnessy by disclosing the step of 'collectively manipulating the converted symbolic representations prior to the step of running the program on the converted symbolic representations' by suggesting that a string representation can be used to sort and index the converted dates such that the dates appear in chronological order (page 945 et seq).
> As to claims 36-39, Ohms, Hazama. and Booth disclose the invention as discussed in the rejection of claim 34. Booth further complements Ohms and Hazama by disclosing the step of 'collectively sorting the converted symbolic representations prior to the step of running the program on the converted symbolic representations' by suggesting that a string representation can be used to sort and index the converted dates such that the dates appear in chronological order (page 945 et seq).
> As to claims 40-45 and 48-59, Shaughnessy and Booth disclose the invention as discussed in the rejection of claim 49 Booth further complements Shaughnessy by disclosing the step of 'collectively manipulating the converted symbolic representations' by suggesting that a string, representation can be used to sort and index the converted dates such that the dates appear in chronological order (page 945 et seq), whereby said dates, contained in a field of the database, are sorted accordingly in a different field of the database (page 839-40 et seq)
> As to, claims 40-45 and 48-59, Ohms, Hazama and Booth disclose the invention as discussed in the rejection of claim 34. Booth further complements Ohms and Hazama by disclosing - the step of 'collectively sorting the converted symbolic representations according to a different data field contained in the database from the at least one date field, prior to the step of running the program on the converted symbolic representations' by suggesting that a string representation can be used to sort and index the converted dates such that the date's appear in chronological order (page 945 et seq), whereby said dates, contained in a field of

> the database, are sorted accordingly in a different field of the database (page 839-40 et seq).
>
> As to claims 46-47, Shaughnessy and Booth disclose the invention as discussed -in the rejection of claim 35. Shaughnessy further discloses the step of 'converting at least a substantial portion of each of the plurality of symbolic representations of dates in the at least one date field and repeating this step until each of the date data entries in the at least one date field is converted into the format that does not have the ambiguity' by 'converting the current date stored in the database field from an ambiguous six digit format (YYMMOD) ---into an unambiguous 8-digit format (CCYYMMDD), wherein the century for the date is specified (col. 5, lines 48-50.et seq).
>
> As to claims 46-47, Ohms, Hazama, and Booth discloses the invention as discussed in the rejection of claim 34. Ohms further discloses the step of 'converting at least a substantial portion of each of the plurality of symbolic representations of dates in the at least one date field and repeating this step until each of the date data entries in the at least one date field is converted into the format that does not have the ambiguity' by converting the current date stored in the database field from an. ambiguous six digit format (YYMMDD) into an unambiguous 8-digit format (CCYYMMDD), wherein the century for the date is specified (p 248, right hand column et seq).

With respect to claim 34, neither Shaughnessy, nor Ohms, nor Booth nor Hazawa, nor any combination of these references discloses or suggests the claimed invention as recited in Claim 34. At a minimum, these references separately or collectively fail to teach or suggest "converting each of the symbolic representations of dates stored in ... the database ... ." Further, they do not teach or suggest doing the reformatting "by windowing ... each of the respective dates as stored ... without the addition of any new data field to the database ... ." In addition there is no teaching or suggestion of "running a program collectively on each of the converted symbolic representations of each of the respective dates ... ."

Ohms and Booth, utilizing Lillian and integer date formats, respectively, do not reformat dates in the database at all, and do not even have the Y2K ambiguity problem addressed by the claimed invention. Shaughnessy and Hazama encounter a date datum and call a subroutine or module to process the single date, or at most two dates, for each given execution of the subroutine/module for resolution of the Y2K ambiguity problem. This is not the claimed invention, as distinguished from these references by at least the recitations noted above.

With respect to claim 35, in addition to the arguments made with respect to claim 34, these references do not separately or collectively teach or suggest "opening the database prior to the step of converting" in the process as recited in claim 34.

Similarly with respect to claims 36 and 37, in addition to the arguments made with respect to claims 34 and 35, these references do not separately or collectively teach or suggest "collectively sorting the converted symbolic representations prior to the step of running the program" in the processes as recited in claims 34 and 35.

Similarly with respect to claims 38 and 39, in addition to the arguments made with respect to claims 34 and 35, these references do not separately or collectively teach or suggest "collectively manipulating the converted symbolic representations prior to the step of running the program" in the processes as recited in claims 34 and 35.

Similarly with respect to claims 40 and 41, in addition to the arguments made with respect to claims 34 and 35, these references do not separately or collectively teach or suggest "collectively sorting the converted symbolic representations according to a different data field contained in the database … ." Neither do they teach or suggest doing this "prior to the step of running the program on the converted symbolic representations" as recited in claims 40 and 41.

Similarly with respect to claims 42 and 43, in addition to the arguments made with respect to claims 34 and 35, these references do not separately or collectively teach or suggest "collectively manipulating the converted symbolic representations according to a different data field contained in the database … ." Neither do they teach or suggest doing this "prior to the step of running the program on the converted symbolic representations" as recited in claims 42 and 43.

Similarly with respect to claims 44 and 45, in addition to the arguments made with respect to claims 34 and 35, these references do not separately or collectively teach or suggest the process as claimed in claims 34 and 35 wherein in addition "the program performs an operation which manipulates the data in a data field associated with the at least one date data field of the database according to the converted symbolic representation of the date" as recited in claims 44 and 45.

Similarly with respect to claims 46 and 47, in addition to the arguments made with respect to claims 34 and 35, these references do not separately or collectively teach or suggest the process as claimed in claims 34 and 35 wherein in addition "the step of converting includes converting at least a substantial portion of each of the plurality of symbolic representations of dates … and repeating this step until each of the date data entries in the at least one date data field is converted" as recited in claims 46 and 47.

Similarly with respect to claims 48 and 49, in addition to the arguments made with respect to claims 46 and 47, these references do not separately or collectively teach or suggest "collectively sorting the converted symbolic representations prior to the step of running the program" as recited in claims 48 and 49.

Similarly with respect to claims 50 and 51, in addition to the arguments made with respect to claims 46 and 47, these references do not separately or collectively teach or suggest "collectively manipulating the converted symbolic representations" as recited in claims 50 and 51.

Similarly with respect to claims 52 and 53, in addition to the arguments made with respect to claims 46 and 47, these references do not separately or collectively teach or suggest "collectively manipulating the converted symbolic representations according to a different data field … prior to the step of running the program" as recited in claims 52 and 53.

Similarly with respect to claims 54 (as amended) and 55, in addition to the arguments made with respect to claims 52 and 53, these references do not separately or collectively teach or suggest "collectively manipulating the converted symbolic representations" as recited in claims 54 and 55.

Similarly with respect to claims 56 - 59, in addition to the arguments made with respect to claims 52 - 55, these references do not separately or collectively teach or suggest a process "wherein the program performs an operation which manipulated the data in the date data field … according to the converted symbolic representation of the date" as recited in claims 56 -59.

Claims 60-65 stand rejected under 35 U.S.C.§103(a) as being unpatentable over Shaughnessy in view of Booth, or in the alternative, over Shaughnessy in view of Hazama, and further in view of Booth.

Claims 60-65 stand rejected under 35 U.S.C.§103(a) as being unpatentable over Ohms in view of Hazama, further in view of Booth.

The Examiner has applied the references similarly to the above applications.

With respect to claim 60, neither Ohms, Shaughnessy, Booth or Hazama, separately or collectively teaches or suggests "converting each of the symbolic representations of dates stored in the at least one date field of the database to a symbolic representation of each of the respective dates that does not create the ambiguity," as recited in claim 60. In addition, they do not separately or collectively teach or suggest doing this "without modifying any of the symbolic representations of dates in the at least one date field of the database for purposes of such windowing and converting," as recited in claim 60. Neither do they separately or collectively teach or suggest "running a program on each of the converted symbolic representations of each of the respective dates ...according to the dates represented by the converted symbolic representations," as recited in claim 60. In addition they do not separately or collectively teach or suggest doing so "separately from the date data symbolic representations of dates contained in the at least one date field," as recited in claim 60.

With respect to claim 61, the same arguments made with respect to claim 60 also apply to claim 61.

With respect to claim 62, the same arguments made with respect to claim 60 also apply to claim 62 and in addition, these references do not teach or suggest the step of "converting" including "without the addition of any new data field to the database for purposes of such windowing and converting," as recited in claim 62. Neither do they separately or collectively teach or suggest "storing the converted symbolic representations separate from the at least one date field of the database," as recited in claim 62. Neither do they teach or suggest "running a program on the stored converted symbolic representations," as claimed in claim 62.

With respect to claim 63, the same arguments that apply to claim 62 also apply to claim 63.

With respect to claims 64 and 65, the same arguments as applied to claims 62 and 63 apply to claims 64 and 65 with the exception that the claimed step of "converting" includes "without modifying any of the symbolic representations of date in the at least one date field of the database for purposes of such windowing and converting," which is not taught or suggested by these references separately or collectively in a process as defined by the recitations of claims 64 and 65, nor with the additional step of "storing the converted symbolic representations separate from the at least one date field in the database," as recited in claims 64 and 65.

Claims 66-69 stand rejected under 35 U.S.C.§102(e) as anticipated by Shaughnessy or, in the alternative, under 35 U.S.C.§103 (a) as obvious over Shaughnessy in view of Hazama.

Claim 66-67 stand rejected under 35 U.S.C.§103(a) as being unpatentable over Ohms in view of Hazama, further in view of Booth.

Claim 68 stands rejected under 35 U.S.C.§103(a) as being unpatentable over Ohms in view of Hazama.

Claim 69 stands rejected under 35 U.S.C.§103(a) as being unpatentable over Ohms in view of Hazama, further in view of Booth.

With respect to claims 66 and 67, these references do not teach or suggest, separately or collectively "reformatting the symbolic representation of each symbolic representation of a date in a portion of the at least one date field in the database, without the addition of any new date field to the database …; and repeating the step of reformatting until each symbolic representation of a date in the at least one date field has been reformatted in order to facilitate collectively further processing the reformatted symbolic representations," as recited in claims 66 and 67.

With respect to claim 68, these references do not teach or suggest, separately or collectively, "reformatting the symbolic representation of each symbolic representation of a date in at least one date field in the database, without the addition of any new date field to the database … in order to facilitate processing of the reformatted symbolic representations … by running a program on the reformatted symbolic representation of each of the dates" as recited in claim 68.

With respect to claim 69, the same arguments as made with respect to claim 68 apply to claim 69 an in addition, the claim recites "sorting the reformatted symbolic representations … and running a program on the reformatted symbolic representations of each of the dates," which is not taught or suggested by these references, separately or collectively.

Claims 70-71 stand rejected under 35 U.S.C.§103(a) as being unpatentable over Shaughnessy in view of Booth, or in the alternative, over Shaughnessy in view of Hazama, and further in view of Booth.

Claims 72-73 stand rejected under 35 U.S.C.§102(e) as anticipated by Shaughnessy or, in the alternative, under 35 U.S.C.§103(a) as obvious over Shaughnessy in view of Hazama.

Claims 74-76 stand rejected under 35 U.S.C.§103(a) as being unpatentable over Shaughnessy in view of Booth, or in the alternative, over Shaughnessy in view of Hazama, and further in view of Booth.

Claims 70-71 stand rejected under 35 U.S.C.§103(a) as being unpatentable over Ohms in view of Hazama, further in view off Booth.

Claims 72-73 stand rejected under 35 U.S.C.§103(a) as being unpatentable over Ohms in view of Hazama.

Claims 74-76 stand rejected under 35 U.S.C.§103(a) as being unpatentable over Ohms in view of Hazama, further in view off Booth.

The Examiner has applied the references as above.

With respect to claims 70, the same arguments made with respect to claim 61 apply to claim 70, except that claim 70 recites both "without the addition of any new data field" and "without modifying any of the symbolic representations of dates in the at least one date field … ."

With respect to claims 71, the same arguments made with respect to claim 62 apply to claim 71.

With respect to claim 72, neither Ohms, Shaughnessy, Booth nor Hazama, separately or collectively, teaches or suggests "reformatting the symbolic representation of each symbolic representation of a date in the database with the values … prior to collectively further processing information contained within the database associated with the respective dates," as recited in claim 72.

With respect to claim 73, neither Ohms, Shaughnessy, Booth nor Hazama, separately or collectively, teaches or suggests "determining a century designator … for each symbolic representation of a date in the database" and "reformatting the symbolic representation of the date … to facilitate further processing of the dates," as recited in claim 73.

With respect to claim 74, neither Ohms, Shaughnessy, Booth nor Hazama, separately or collectively, teaches or suggests "determining a century designator … for each symbolic representation of a date in the database" and "reformatting the symbolic representation of the date … to facilitate further processing of the dates," and "sorting the dates in the [reformatted] form," as recited in claim 74.

With respect to claim 75, neither Ohms, Shaughnessy, Booth nor Hazama, separately or collectively, teaches or suggests "determining a century designator … for each symbolic representation of a date in the database" and "reformatting the symbolic representation of each symbolic representation of a date in the database, without the addition of any new data field to the database … to facilitate further processing of the reformatted symbolic representations of each of the symbolic representations of each of the dates," as recited in claim 75.

With respect to claim 76, the same arguments that applied to claim 75 apply to claim 76 and in addition the claim recited "sorting the dates in the [reformatted] form … ."

For the above stated reasons, claims 1-76, including claim 54 as amended, should be allowed and the Examiner is respectfully requested to allow claims 1-76 and to issue a Notice of Allowance regarding these claims.

Respectfully submitted,

William C. Cray
Reg.No. 27,627